

Лабораторная работа №2

Моделирование и обработка сигналов в пакете MATLAB

Цель работы: ознакомиться с методикой моделирования и обработки периодических дискретных сигналов с помощью пакета MATLAB.

1. Теоретические сведения.

Принцип построения алгоритмов БПФ

Рассмотрим выражение для дискретного преобразования Фурье:

$$S(k) = \sum_{n=0}^{N-1} s(n) \cdot \exp\left(-j \cdot \frac{2\pi}{N} \cdot n \cdot k\right), k = 0 \dots N - 1. \quad (1)$$

ДПФ N отсчетов сигнала $s(n)$, $n = 0 \dots N - 1$, (в общем случае комплексного) ставит в соответствие N комплексных спектральных отсчетов $S(k)$, $k = 0 \dots N - 1$. Для вычисления одного спектрального отсчета требуется N операций комплексного умножения и сложения. Таким образом, вычислительная сложность алгоритма ДПФ составляет N^2 операций комплексного умножения и сложения.

Поскольку сложность алгоритма растет квадратично относительно размера входного сигнала, можно достичь существенного ускорения вычисления, если нам удастся свести расчет N – точечного ДПФ к двум $\frac{N}{2}$ – точечным ДПФ, как это показано на рисунке 1.

Замена одного N – точечного ДПФ двумя $\frac{N}{2}$ – точечными ДПФ приведет к уменьшению количества операций в 2 раза, но дополнительно требуются операции разделения последовательности на две и объединение двух $\frac{N}{2}$ – точечных ДПФ в одно N – точечное.

При этом каждое из $\frac{N}{2}$ – точечных ДПФ также можно вычислить путем замены $\frac{N}{2}$ – точечного ДПФ на два $\frac{N}{4}$ – точечных, которые, в свою очередь, можно рассчитать через $\frac{N}{8}$ – точечные ДПФ. Эту рекурсию можно продолжать, пока возможно разбить входную последовательность на две.

В нашем случае, если $N = 2^L$, L — это положительное целое, мы можем разделить последовательность пополам L раз. Для $N = 8$ ($L = 3$) такое разделение представлено на рисунке 2.

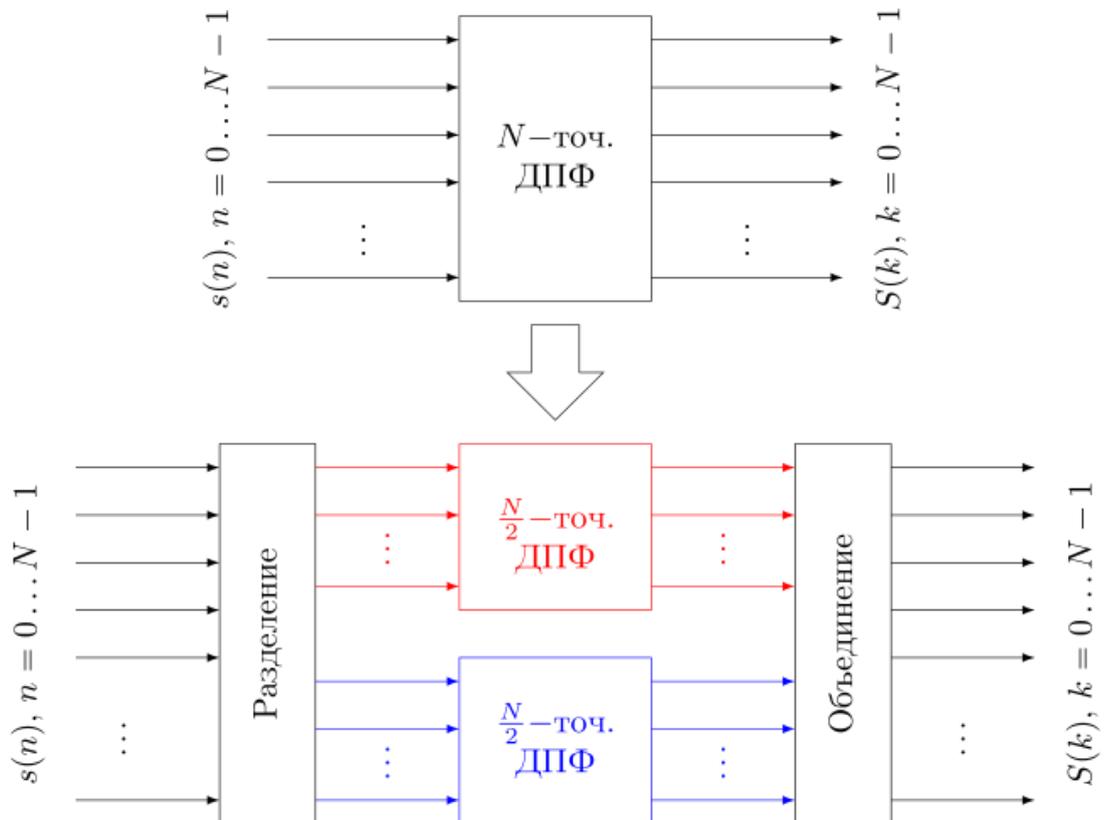


Рисунок 1. Замена N — точечного ДПФ двумя $\frac{N}{2}$ — точечными ДПФ

Алгоритмы БПФ, которые используют выборки длиной $N = 2^L$, называются «алгоритмами БПФ по основанию 2». Данные алгоритмы получили наибольшее распространение из-за их высокой эффективности и относительной простоты программной реализации.

Мы рассмотрим два способа разделения — объединения: прореживание по времени и прореживание по частоте.

Обратное быстрое преобразование Фурье

Эффективный алгоритм вычисления прямого БПФ можно использовать и для обратного преобразования. Обратим внимание, что комплексные экспоненты в выражениях для прямого и обратного ДПФ являются комплексно-сопряженными:

$$\exp\left(j \cdot \frac{2\pi}{N} \cdot N \cdot k\right) = \left(\exp\left(-j \cdot \frac{2\pi}{N} \cdot N \cdot k\right)\right)^*, \quad (2)$$

где $(\cdot)^*$ - оператор комплексного сопряжения.

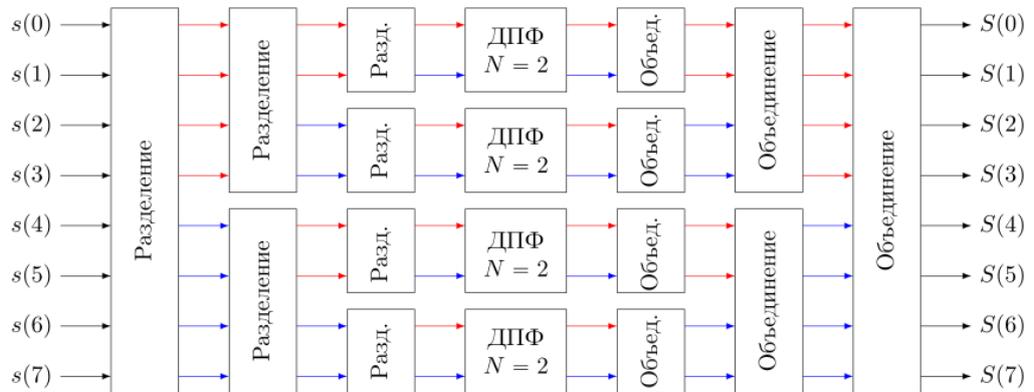


Рисунок 2. Разделение и объединение последовательности для $N = 8$

Нетрудно показать, что для двух комплексных чисел $x = a + j \cdot b$ и $y = c + j \cdot d$ справедливо следующее равенство:

$$x \cdot y^* = (x^* \cdot y)^*. \quad (3)$$

Применительно для выражения ОДПФ можно записать:

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) \cdot \left(\exp\left(-j \cdot \frac{2\pi}{N} \cdot n \cdot k\right)\right)^* = \frac{1}{N} \left(\sum_{k=0}^{N-1} S(k) \cdot \exp\left(-j \cdot \frac{2\pi}{N} \cdot n \cdot k\right)\right)^*. \quad (4)$$

Таким образом, берется комплексно-сопряженный спектр $S^*(k)$, выполняется прямое ДПФ и результат подвергается комплексному сопряжению. Вычисление ОДПФ при использовании ДПФ приведено рисунке 3.

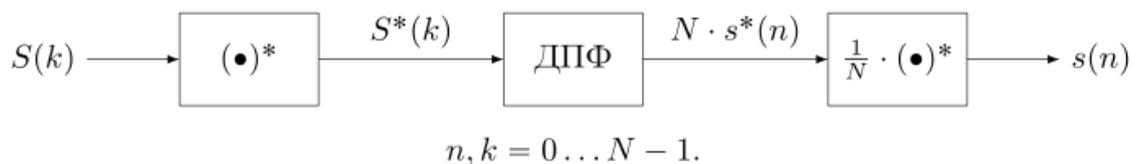


Рисунок 3. Вычисление обратного БПФ

Если вместо ДПФ использовать БПФ, то получим обратное быстрое преобразование Фурье (ОБПФ). При этом для выполнения комплексного

сопряжения необходимо лишь поменять знак перед мнимой частью спектра до вызова функции БПФ и результата после БПФ.

Реализации в MATLAB

Генерировать сигналы в MATLAB можно тремя способами:

- в диалоговом режиме, с помощью последовательности команд в командном окне;
- в автоматическом режиме, путем создания и запуска на выполнение m-скрипта;
- в автоматическом режиме, путем создания и вызова m-функции.

Генерирование сигналов в диалоговом режиме.

Этот способ наиболее трудоемок, поскольку требует каждую команду набирать с клавиатуры в командном окне. Чтобы повысить производительность труда, можно всю последовательность команд предварительно набрать в любом текстовом редакторе (обычно это Notebook или Word), а затем, скопировав текст в буферную память (Clipboard), вставить его в командное окно. Недостаток этого способа в том, что необходимо одновременно держать активными две программы – MATLAB и текстовый редактор. Достоинство данного способа проявляется тогда, когда работу в MATLAB производят, следуя некоей инструкции, в которой теоретические сведения чередуются с практическими заданиями в виде фрагментов текстов m-скриптов.

Например, так выглядит в текстовом редакторе последовательность команд генерирования N отсчетов тонального сигнала амплитудой A , частотой f_0 , начальной фазой Fi_0 , с частотой дискретизации F_s :

```
% гармонический сигнал
% параметры сигнала
A=1;
f0=100;
Fi0=pi/2;
Fs=1000;
N=20;
t=(0:N-1)/Fs; % моменты времени
```

```

s=A*sin(2*pi*f0*t+Fi0); % вычисление отсчетов
%графики
plot(t,s) % вывод графика
title('Гармонический сигнал') % заголовок
% надписи вдоль осей
xlabel('Время, с');
ylabel('Уровень');
grid on % координатная сетка

```

Полученный график отображается в специальном окне с надписью Figure #1 (если это первый строящийся график). График удобно сохранять путем экспорта в экономном формате *.jpg (рис.4).

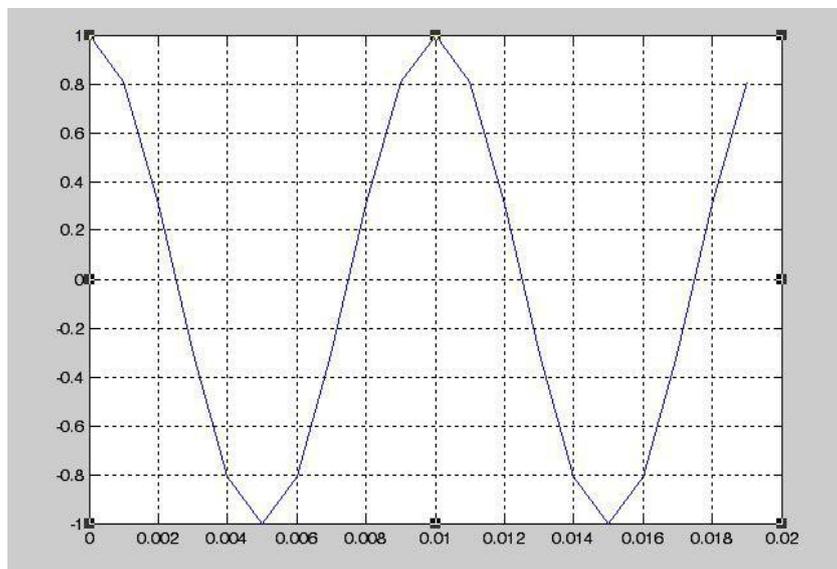


Рисунок 4. График сигнала тональной частоты

Примечание: при использовании символов кириллицы в тексте команд (комментарии, заголовки и т.п.) следует учитывать особенности «отношения» каждой конкретной версии MATLAB к кириллице. Так, в MATLAB версии 6.1 нельзя употреблять строчную букву «я» – вместо нее следует писать прописную букву «Я». Именно по этой причине в тексте на рис. 2 вместо «Время» получилось «Время». Впрочем, эту надпись можно отредактировать (кнопка со стрелкой Edit Plot в графическом окне) перед тем, как сохранять рисунок на диске.

Генерирование сигналов путем создания m-скрипта.

Данный способ отличается тем, что все команды набираются в специальном окне редактора m-файлов (рис.5).

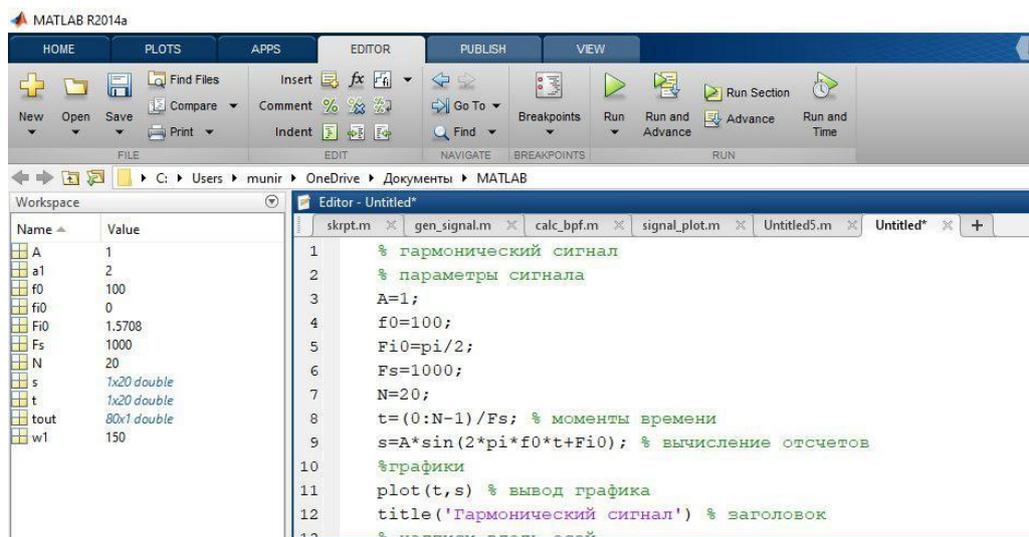


Рисунок 5. Набор команд в окне редактора m-файлов

Данный способ удобен тем, что вместо сторонних программных продуктов используется собственный инструментарий MATLAB, специализированный для написания и отладки m-скриптов.

Генерирование сигналов путем создания m-функции.

Данный способ отличается тем, что входные данные записывают как аргумент некоей функции $y = f(x)$, а выходные – как значение этой функции. Удобство в том, что символьные обозначения данных могут отличаться от обозначений, используемых в теле функции. Более того, числовые значения входных данных можно просто задавать в наименовании вызываемой функции. Последнее обстоятельство продемонстрируем на примере.

Создадим подпрограмму – m-скрипт `ton.m` вида:

`% скрипт ton`

`s=A*sin(2*pi*f0*t+Fi0); % вычисление отсчетов сигнала`

Команду выполнения этого скрипта нужно «окружить» командами подготовки входных данных и вывода выходных данных:

`A=1; f0=100; Fi0=pi/2; Fs=1000; N=20; % параметры сигнала`

`t=(0:N-1)/Fs; % моменты времени`

`ton; % вычисление отсчетов`

`%график`

`plot(t,s) % вывод графика`

`title('Гармонический сигнал') % заголовок`

```
xlabel('Время, с'); ylabel('Уровень'); % надписи вдоль осей
grid on % координатная сетка
```

Очевидно, обозначения входных и выходных данных вызывающей программы должны совпадать с обозначениями соответствующих данных вызываемой подпрограммы.

Теперь поступим по-иному – напишем и сохраним m-функцию под именем ton_sig.m:

```
%-----функция ton_sig.m -----
% [s,t]=ton_sig(B,f1,Fi1,Fs,N1)
%-----
% генерирование гармонического сигнала
%  $y = B * \sin(2*\pi*f1*x + Fi1)$ ,
% B - амплитуда;
% N1 - количество отсчетов сигнала;
% f1 - частота;
% Fs - частота дискретизации;
% Fi1 - начальная фаза сигнала
%-----
function [y,x] = ton_sig( B, f1, Fi1, Fs, N1 )
%-----
x = (0:N1-1)/Fs; % моменты времени
y = B * sin( 2*pi*f1*x + Fi1 );
%----- конец функции ton_sig.m -----
```

Теперь m-скрипт генерирования того же отрезка косинусоиды будет выглядеть так:

```
% гармонический сигнал
[s,t]=ton_sig(1,100,pi/2,1000,20) % вычисление отсчетов сигнала
plot(t,s) % вывод графика
title('Гармонический сигнал') % заголовок
xlabel('Время, с'); ylabel('Уровень'); % надписи вдоль осей
grid on % координатная сетка
```

Как видим, теперь числовые значения входных данных задаются как аргументы m-функции ton_sig.m. Выходные данные функции используются для построения графика.

Очевидно, применение m-функций предпочтительно в том случае, когда алгоритм формирования значений функции достаточно сложный, содержится много команд и обращений к разнообразным библиотечным функциям с непростым синтаксисом.

Очевиден и недостаток m-функций – необходимо помнить их синтаксис. Впрочем, получить нужную информацию можно, если в командном окне задать команду help:

```
>> help ton_sig
```

В результате на мониторе отобразится комментарий, с которого начинается m-функция. Для приведенного выше примера текст помощи имеет следующий вид:

```
%-----функция ton_sig.m -----  
% [s,t]=ton_sig(B,f1,Fi1,Fs,N1)  
%-----  
% генерирование гармонического сигнала  
% y = B * sin(2*pi*f1*x + Fi1),  
% B - амплитуда;  
% N1 - количество отсчетов сигнала;  
% f1 - частота;  
% Fs - частота дискретизации;  
% Fi1 - начальная фаза сигнала  
%-----
```

Таким образом, очевиден вывод: очень важно при программировании m-функций снабжать их качественным и подробным комментарием.

2. Порядок выполнения работы

Часть 1.

1. Получить у преподавателя номер варианта
2. Скачать с Google диска Excel файл записи отсчетов дискретного сигнала.
3. Перенести отсчеты сигнала из Excel файла в MATLAB. Частота дискретизации с которой записан сигнал $F_s = 2^{10}$.
4. Построить график сигнала.
5. Самостоятельно ознакомиться с функцией fft в справке MATLAB.

6. Используя БПФ получить амплитудный и фазовый спектры сигнала.

7. Построить графики амплитудного и фазового спектра.

8. Примерно определить амплитуды и частоты гармонических сигналов входящих в сигнал по варианту. Записать их в таблицу.

Часть 2.

1. Создать m-файл.

2. Используя таблицу 1 задать значения переменных, соответствующие значениям двух гармонических сигналов по варианту.

3. Сгенерировать сигнал, представляющий собой сумму двух синусоид и гауссовского шума с нулевым математическим ожиданием и СКО равным единице.

4. Используя БПФ получить амплитудный спектр смеси гармонических сигналов с шумом.

5. Оценить значения амплитуды и частоты по амплитудному спектру.

6. Увеличивать СКО шума до 6 с шагом 0.5. Оценивать значения амплитуды и частоты.

7. Построить график зависимости ошибки определения амплитуды сигнала в зависимости от СКО шума.

8. Определить эмпирически значение СКО шума, при котором визуально не получится определить гармонические составляющие смеси сигнала с шумом. Продемонстрировать данную ситуацию на графике.

3. Требования к оформлению отчета

1. Оформление по ГОСТ 7.32-2017

2. Титульный лист

3. Цель работы

4. Листинг программы

5. Графики и таблицы

6. Выводы

Таблица 1 – варианты заданий

№	1	2	3	4	5	6	7	8	9	10
A_1	2	3	4	5	6	5	4	3	2	3
ω_1	100	150	200	250	300	100	150	200	250	300
φ_1										
A_2	6	4	5	3	2	2	3	2	6	2
ω_2	300	250	150	200	100	300	250	150	200	100
φ_2										

№	11	12	13	14	15	16	17	18	19	20
A_1	200	300	400	500	600	500	400	300	200	300
ω_1	60	50	40	30	20	60	50	40	30	20
φ_1										
A_2	500	400	300	200	300	400	300	200	400	200
ω_2	40	60	50	60	50	40	30	50	20	30
φ_2										

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1.Ревинская О.Г. Основы программирования в Matlab:учеб.пособие.-СПб.:БХВ-Петербург,2016.-208 с.:ил.-(Учебное пособие)

2.URL:https://www.mathworks.com/help/coder/ref/matlabcoder-app.html?searchHighlight=Projects&s_tid=doc_srchttitle.(дата обращения: 4.07.2019)

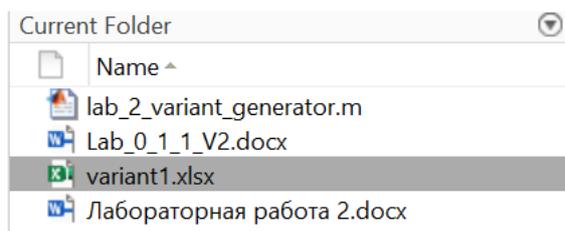
4.URL:https://it.mathworks.com/help/images/code-generation-with-cell-detection.html?action=changeCountry&s_tid=gn_loc_drop.(дата обращения 4.07.2019)

4.Ануфриев И.Е. Самоучитель MATLAB 5.3/6.x / И.Е. Ануфриев. СПб.: БХВ-Петербург, 2003. 736 с.

5.Дьяконов В.П. MATLAB 6.5 SPI/7 + Simulink 5/6 в математике и моделировании / В.П. Дьяконов. М.: СОЛОН-Пресс, 2005. 576 с. (Серия «Библиотека профессионала»).

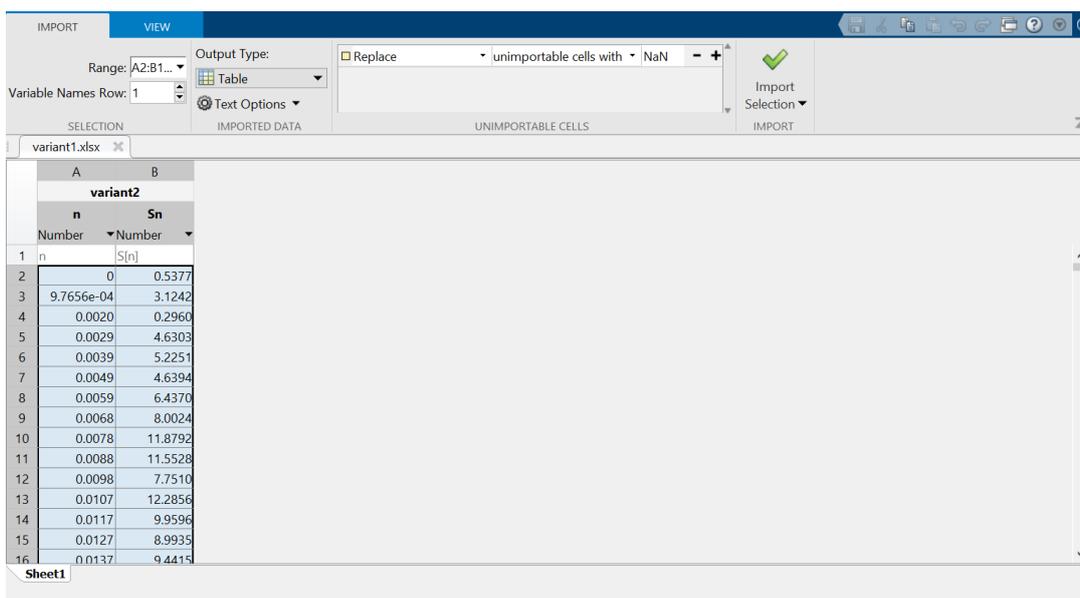
Приложение 1. Перенос данных из Excel в MATLAB

1. Поместить файл variant.xlsx в рабочую папку MATLAB.

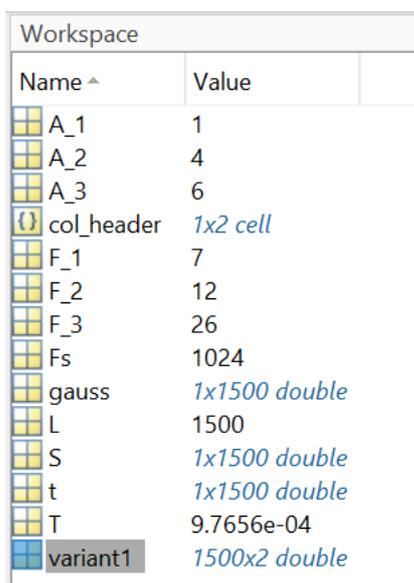


2. 2 раза нажать левой кнопкой мыши по файлу variant.xlsx.

Появится окно следующего вида:



3. В графе «Output Type» выбрать Numeric matrix.
4. Нажать Import Selection.
5. В окне Workspace должна появиться матрица размером 1500x2



The screenshot shows the MATLAB Workspace window. It contains a table with the following data:

Name	Value
A_1	1
A_2	4
A_3	6
col_header	1x2 cell
F_1	7
F_2	12
F_3	26
Fs	1024
gauss	1x1500 double
L	1500
S	1x1500 double
t	1x1500 double
T	9.7656e-04
variant1	1500x2 double

6. Записать отсчеты в отдельные вектор столбцы можно следующими командами:

```
x = variant1(:,1);
```

```
y = variant1(:,2);
```

где x – время, в которое был записан отсчет, y – значение сигнала.

7. Готово. Теперь над сигналом можно осуществлять необходимые манипуляции.